
Dual Contouring for Domains with Topology Ambiguity

Jin Qian and Yongjie Zhang *

Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA.

Summary. This paper describes an automatic and robust approach to generate quality triangular meshes for complicated domains with topology ambiguity. In previous works, we developed an octree-based Dual Contouring (DC) method to construct triangular meshes for complicated domains. However, topology ambiguity exists and causes non-conformal meshes. In this study, we discuss all possible topology configurations and develop an extension of DC which guarantees the correct topology. We first generate one base mesh with the previous DC method. Then we analyze all the octree leaf cells and categorize them into 31 topology groups. In order to discriminate these cells, we compute the values of their face and body saddle points based on a tri-linear representation inside the cells. Knowing the correct categorization, we are able to modify the base mesh and introduce more minimizer points within the same cell. With these minimizer points we update the mesh connectivities to preserve the correct topology. Finally we use a Laplacian smoothing technique to improve the mesh quality. Our main contribution is the topology categorization and mesh modification. We have applied our algorithm to three complicated domains and obtained good results.

Key words: Dual Contouring, topology ambiguity, triangular mesh, tri-linear representation, saddle points

1 Introduction

Accurate representation of an iso-surface is one important problem in scientific visualization and mesh generation. Given 3D scanned images, we aim to generate quality triangular surfaces with correct topology. The Marching Cubes (MC) algorithm [14] visits each cell in the volume and performs local triangulation based on the sign configuration of the eight grid points. Accelerated algorithms [2, 21] were further developed to reduce the running time by avoiding visiting unnecessary cells. The iso-surface inside the cubic cells may have complicated shapes or topology ambiguities. To handle them, the function values of the face and body saddles in the cell are used to decide the correct topology and to generate consistent triangulation [15, 13]. Main drawbacks of MC include uniform and large-size mesh, badly

*Corresponding author. Telephone: (412) 268-5332; Fax: (412) 268-3348; Email: jessicaz@andrew.cmu.edu

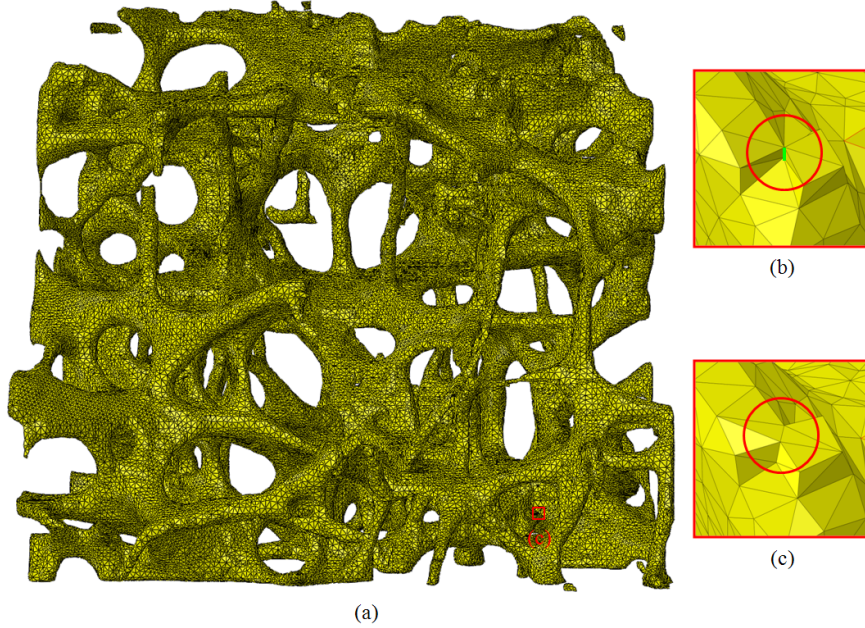


Fig. 1. (a) Triangular mesh of a trabecular bone structure with complicated topology. (b) and (c) show zoom-in details of a local region with topology ambiguity (see the red circle). The green line in (b) denotes one non-manifold edge, which is resolved in (c).

shaped triangles as well as the loss of sharp features. To generate an adaptive iso-surface, people developed ways to triangulate cells with different levels. However, when adjacent cells have different resolution levels, cracks are introduced and a fan of triangles have to be inserted around the gravity center of the coarse triangles [20]. Octree based Dual Contouring (DC) method [10] combines SurfaceNets [8] and the extended Marching Cubes [12] algorithms, and it is able to generate adaptive iso-surfaces with good aspect ratios and sharp feature preservation. Despite being adaptive and feature-preserving, DC has the drawback that one cell can contain only one minimizer, leaving possible non-manifold meshes. To address this deficiency, the vertex clustering algorithm [22, 19] together with topology constraints [17, 3, 11] was developed. However, for datasets with very complicated topology (see the trabecular bone structure in Fig. 1), the existing DC methods fail to recognize all the topology ambiguities and may lead to non-conformal meshes.

To distinguish the topology ambiguities, we need to study the function properties inside the cubic cell. Since the function value is only given at the eight grid points of each cell, we model the interior region with a tri-linear interpolation. Depending on the function values of these eight grid points, there are 14 unique configurations for one cell. For some of these configurations, topology ambiguities arise. These ambiguities are either on the face of the cell or interior to the cell. To discriminate them, we compute the function values at body and face saddle points [15]. With respect to the values at the saddle points, we decide whether the iso-surface forms a

tunnel linking the two grid points or it is separated into several parts. With all these ambiguities resolved, we develop a new algorithm that modifies the mesh generated using the standard DC cell by cell. In this algorithm, multiple minimizers may be introduced to one cell, eliminating the drawback that the DC method has only one minimizer within one cell. To improve the mesh quality, we relocate the vertex positions via a Laplacian smoothing technique [25]. Our algorithm keeps the mesh conformal across cell faces and attains quality meshes with correct topology.

The remainder of this paper is organized as follows: Section 2 reviews related previous work. Section 3 talks about the standard DC method. Section 4 and 5 discuss the resolving and triangulation of 2D and 3D ambiguities. Section 6 presents results and discussion, and finally Section 7 draws conclusions.

2 Previous Work

Marching Cubes. As one of the most popular iso-contouring techniques, the MC algorithm [14] classifies cubic cells into 256 configurations, depending on whether the eight vertices are positive or negative. After considering symmetry and complementary, the 256 cases can be reduced to 14 unique ones. If the two endpoints of any edge have different signs, then the edge is intersected by the iso-surface. The intersection point can be estimated via a linear interpolation. For each of the 14 cases, the approximation of the iso-surface can be created by the triangulation of multiple intersection points. These configurations are incorporated into a lookup table, and each entry in the table contains a triangulation pattern. Then the MC visits one cell at a time until all the cells are treated.

The MC is straight-forward and easy to implement; however, it has several drawbacks. The main problem of MC is that it requires uniform cell structure, which may lead to huge mesh size. Meanwhile, the vertices in MC are restricted on the cell edges, and this introduces many elements with small angles. Furthermore, sharp features in the data are not well preserved. Another drawback is that there is a possibility for discrepancy in the connection of the shared face of two adjacent cells, which are caused by un-categorized ambiguities.

Extended Marching Cubes. The MC has been the focus of much further research to improve its quality of iso-surface representation [13]. To improve the mesh quality, additional information in the volume, such as surface normals, are utilized to reduce sharp angles [12]. To make the iso-surface adaptive, people developed ways to triangulate over cells with different levels. However, when the resolution levels of adjacent cells are different, there will be cracks and a fan of triangles needs to be inserted around the gravity center of the coarse triangle [20]. To handle the topology ambiguities, face ambiguity [16, 5] and interior ambiguity [15, 4] have been discussed. In order to distinguish these ambiguities, a strategy based on saddle points, where the first partial derivative of the tri-linear function equals to zero, was proposed. After considering all the ambiguities, an enhanced look-up table consisting of 31 cases were built [13]. These improved MC methods provide better quality meshes and handle ambiguities; however, they still lack the capability to mesh large datasets adaptively.

Dual Contouring. The Dual Contouring (DC) method [10, 24], which combines SurfaceNets [8] and the extended Marching Cubes [12] algorithm, generates a dual mesh from an adaptive octree instead of a uniform one. The adaptation of the

octree can be controlled by certain error functions. It was first designed for the triangulation from Hermit data [10] and then extended to tetrahedral and hexahedral mesh generation [24, 23], as well as domains with heterogeneous materials [26]. The DC has several advantages compared with MC. Most importantly it is able to handle large datasets using an adaptive octree and preserves sharp features well. However, a problem of DC is the restriction that each leaf cell only holds one minimizer. Such restriction leads to non-manifold meshes and thus makes the accurate representation of topology ambiguities impossible. As a follow-up, vertex clustering, which inserts multiple minimizers inside one leaf cell, was introduced [22, 19]. It was further applied with topology constraints so that the resulting contours are always 2-manifold [17, 3, 11]. However, these techniques still lack the capability of recognizing all the topology ambiguities.

3 An Overview of Standard Dual Contouring

The DC method [10] was developed based on an adaptive octree data structure, which achieves denser cells along the boundary and coarser cells inside the volume. The procedure begins with splitting the volume data into octree cells iteratively until the mesh adaptation is satisfactory. The mesh adaptation can be controlled adaptively by using a feature sensitive function $F = \sum(|f^{i+1} - f^i| / \nabla f^i)$, where f is a tri-linear interpolation function inside the cell [24]. This feature sensitive error function measures the iso-contour difference between two neighboring octree levels. When the error is below a pre-defined tolerance, the octree splitting ceases. In order to obtain triangles with good quality, we restrict the neighboring level difference to be less than or equal to two. One minimizer was computed for each leaf cell to minimize a predefined Quadratic Error Function (QEF), $QEF(x) = \sum(n_i \cdot (x - p_i))^2$, where p_i and n_i are the position and normal vectors at the intersection points between the iso-surface and octree cell [6, 7].

The next stage is to analyze each sign change edge, whose one endpoint lies inside the internal volume while another lies outside. Since the adaptive octree consists of leaves with different resolution levels, each leaf may have neighbors at different levels, i.e., an edge in one leaf cell may be divided into several edges in its neighboring cells. In deciding which edge should be analyzed, we follow the DC method and always choose the minimal one. These edges are those of leaf cells that do not properly contain an edge of a neighboring leaf. In this adaptive octree, the minimal edge is either shared by four cells, and we connect the four QEF minimizers to form a quadrilateral (quad); or it is shared by three cells and we form a triangle, and we obtain a hybrid mesh consisting of quad and triangular elements. Finally we divide these quads to form a triangular mesh. One splitting diagonal is chosen to achieve better aspect ratios. By now an adaptive triangular mesh is created. Several variants of DC method have been developed as well, which introduce better topology and mesh quality [1, 9, 18, 17, 22].

Despite being adaptive and feature-preserving, one drawback of such DC method is that it fails to handle ambiguities in the original topology and it may introduce non-manifold surfaces, that is, an edge on the contour may be shared by more than two polygons. These surfaces are not only troublesome in visualization but also are problematic in further mesh processing techniques such as smoothing and parameterization. Therefore the ambiguities must be resolved.

4 Modified DC for 2D Ambiguities

Ambiguity arises when the iso-surface can be represented using different approaches and there are multiple ways to connect the minimizers. To capture the ambiguities, we need to categorize all the cells into different cases and then adjust the mesh topology accordingly. We first discuss the simple 2D ambiguity problem followed by the complex 3D problem. Here are some definitions used in the categorization.

Positive grid point: A positive grid point is a grid point whose function value is greater than or equal to the given iso-value.

Negative grid point: A negative grid point is a grid point whose function value is less than the given iso-value.

Joined grid points: If two grid points within the same cell meet the following two criteria, we call them joined grid points: (1) These two points have the same positive/negative sign; (2) there is a path between these two points, along which the interior function value between them does not change sign.

Separated grid points: If two grid points within the same cell only meet the first criterion above, we call these two points separated grid points.

4.1 Resolving 2D Ambiguities

To begin with, we now resolve the simple 2D ambiguities. Since the function values are only given at the four corners of each quadtree cell, here we choose a bi-linear interpolation, $F(\xi, \eta)$, which is written in terms of the function values at four corners F_{ij} ($i, j = 0, 1$):

$$F(\xi, \eta) = F_{00}(1 - \xi)(1 - \eta) + F_{01}(1 - \xi)\eta + F_{10}\xi(1 - \eta) + F_{11}\xi\eta. \quad (1)$$

Depending on the sign of the four grid points, we summarize all 2D possibilities into 4 categories (Cases 0, 1, 2, and 3) with the consideration of symmetry and complementary (reversing positive/negative signs). The meshes generated via the standard DC are shown in Fig. 2. DC generates a series of quads first and then splits them into triangles, therefore when we discuss about triangulation, we only consider how to generate the quads. The cell may only have one positive grid point (Case 1); or it has two neighboring positive grid points (Case 2); otherwise the cell has two positive grid points forming one diagonal: these two grid points can be either joined (Case 3-1) or separated (Case 3-2).

From Case 3-1 and Case 3-2 it is obvious that ambiguity arises when two grid points in one diagonal have the same sign. The standard DC method falls short to recognize this ambiguity and creates non-manifold meshes, as shown in Fig. 2. To discriminate these two possibilities, we find out the saddle point, where the first partial derivative for each direction is zero, and then compute the function value at the saddle point. If that value is positive (greater than or equal to the iso-value), we can tell that the positive zone spreads across the cell so that the two positive grid points are joined; otherwise the positive zone is divided and then the two positive grid points are separated.

After resolving these two ambiguous cases, there are a total of five topology cases for a 2D cell (see Fig. 2).

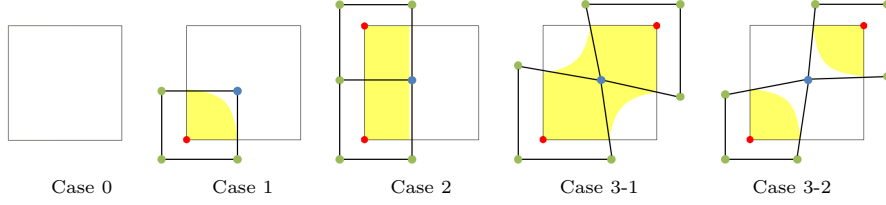


Fig. 2. 2D topology possibilities and the meshing results via the standard DC. The cell has one or two positive grid points (red), one minimizer (blue) and up to six neighboring cell minimizers (green). The iso-surface (yellow) intersects the cell.

Simple configurations: **Case 0, 1, 2** have no ambiguities. These cases can be meshed using the standard DC.

Configuration 3: There are two possible cases:

Case 3.1: Positive grid points are joined inside the cell; and the iso-surface spreads across the cell;

Case 3.2: Positive grid points are separated; and the iso-surface breaks into two pieces inside this cell.

4.2 Handling 2D Ambiguities in Triangulation

With the two ambiguities resolved, we are able to re-mesh the 2D iso-surface. For those simple cases, the traditional DC method is sufficient to represent the correct topology. For the two ambiguous cases, re-meshing is necessary. We first need to insert two new minimizers inside the cell. The minimizers are located on the intersection points of two tangent lines to the iso-surface. Next we check all the quads sharing the original minimizer, and then re-connect the quads using the schemes shown in Fig. 3. The old minimizer is then deleted since no quads connect to it anymore. Till now we obtain a hybrid mesh consisting of quads and pentagons. For a quad we pick one diagonal and divide it into two triangles. For a pentagon, we split it into three triangles.

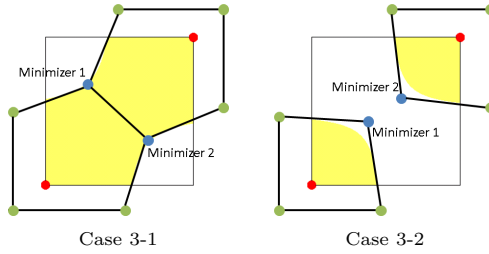


Fig. 3. Resolving 2D ambiguities. Each cell has positive grid points (red) and minimizers (blue); the green nodes denote the minimizers of neighboring cells.

5 Modified DC for 3D Ambiguities

Resolving 3D topology ambiguities within each cell is much more complicated. We first discuss how to resolve these ambiguities and then present the re-meshing techniques to represent the iso-surface accordingly.

5.1 Resolving 3D Ambiguities

In 3D octree cells the function values are given only at the eight grid points, similarly here we choose a tri-linear interpolation, $F(\xi, \eta, \zeta)$, which is written in terms of the function values at the eight grid points, F_{ijk} ($i, j, k = 0, 1$):

$$\begin{aligned} F(\xi, \eta, \zeta) = & F_{000}(1-\xi)(1-\eta)(1-\zeta) \\ & + F_{001}(1-\xi)(1-\eta)\zeta + F_{010}(1-\xi)\eta(1-\zeta) \\ & + F_{011}(1-\xi)\eta\zeta + F_{100}\xi(1-\eta)(1-\zeta) \\ & + F_{101}\xi(1-\eta)\zeta + F_{110}\xi\eta(1-\zeta) \\ & + F_{111}\xi\eta\zeta. \end{aligned} \quad (2)$$

Depending on the sign of the eight grid points, there are $2^8 = 256$ possible configurations for one cubic cell. Considering all the symmetry and complementary, the 256 situations can be reduced to 14 fundamental cases, as shown in Table 1.

Some of these fundamental cases have topology ambiguities, which can be categorized into two groups: **face ambiguity** and **interior ambiguity**. Face ambiguity arises when two positive grid points occupy one cubic face diagonal, as shown in Fig. 4(a-b). The four edges of this face are all sign change edges, and we need to decide how to triangulate this local region. If the mesh is built in an inconsistent way, discrepancies such as a gap may be introduced between adjacent cells.

Meanwhile, there are additional interior ambiguities inside the cubic cell, as shown in Fig. 4(c-d). In these situations the iso-surface can either be several separated surfaces or a tunnel piece passing through the cubic cell. These interior structures generally do not cause immediate discrepancies, but they decide what the topology looks like inside the cell.

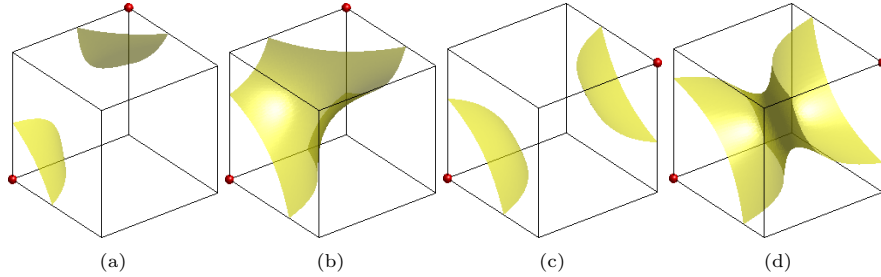
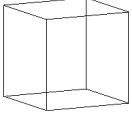
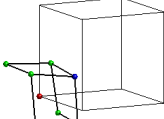
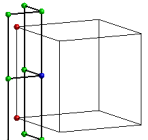
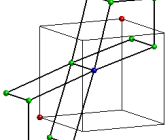
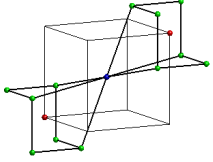
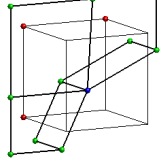
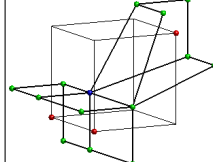
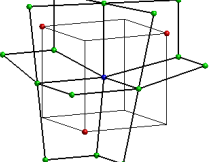
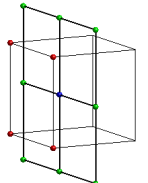
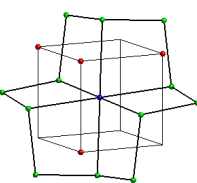
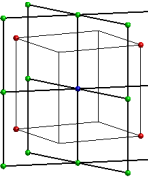
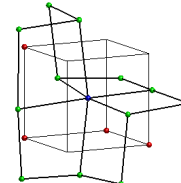
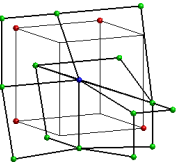
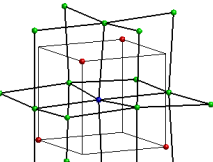


Fig. 4. Example of 3D face and interior ambiguities. The cubic cell has two positive grid points (red) and the iso-surface (yellow) intersects with it. (a-b) One face ambiguity; and (c-d) one interior ambiguity.

Table 1 14 fundamental cases for 3D cubic cells with meshes generated using the standard DC. Each cell has a few positive grid points (red), one minimizer (blue) and several minimizers of adjacent cells (green).

			
Case 0	Case 1	Case 2	Case 3
			
Case 4	Case 5	Case 6	Case 7
			
Case 8	Case 9	Case 10	Case 11
			
Case 12	Case 13		

In order to discriminate one topology from another, we again analyze the function values of the face and interior saddle points. Similar to 2D saddle points, we find the saddle points by setting all the first partial derivatives to be 0. When analyzing face ambiguities, the saddle points are confined to stay on that very face; therefore the tri-linear interpolation degenerates to a bi-linear one, which leads to a single root. On the other hand, for interior ambiguities, the saddle points are set to be inside the cubic cell; so we need to solve a quadratic equation with double roots. If the computed saddle points lie outside the specific surface or the cell, it is ignored. Till now we can compute the function values at these saddle points, and use them to categorize the topology.

After considering all face and interior ambiguities, a comprehensive set of 31 cases are summarized. Since this set is actually a detailed version of the 14 fundamental cases, therefore we use labeling notations inherited from the 14 cases. The labeling of these 31 cases denotes the topologies they belong to: the first number denotes the case number in the original 14 fundamental cases (see Table 1); the second indicates the resolution of a face ambiguity; in the end the last index clarifies an interior ambiguity. These 31 cases are described as follows (see Table 2):

Simple configurations: **Case 0, 1, 2, 5, 8, 9** and **11** have neither face ambiguity nor interior ambiguity. These cases can be meshed using the standard DC method.

Configuration 3: There is only one ambiguous face. After resolving this face, two different cases exist:

Case 3-1: Positive grid points are separated;

Case 3-2: Positive grid points are joined inside the face.

Configuration 4: There is no face ambiguity but one interior ambiguity. Resolving such interior ambiguity leads to two cases:

Case 4-1-1: Positive body diagonal grid points are separated;

Case 4-1-2: Positive grid points are joined inside the cubic cell and thus form a “tunnel” structure.

Configuration 6: There is one ambiguous face; and if the positive grid points are separated on this face, additional interior ambiguity arises: the two positive grid points can be either separated or joined inside the cube. Three cases are possible:

Case 6-1-1: Positive grid points on the ambiguous face are separated and they are not joined inside the cube;

Case 6-1-2: Positive grid points on the ambiguous face are separated and they are joined inside the cube;

Case 6-2: Positive grid points are joined on the ambiguous face.

Configuration 7: There are three adjacent ambiguous faces sharing one negative grid point. If all the positive grid points on the ambiguous faces are separated, the interior ambiguity arises: the common negative grid point can be separated from other negative grid points or joined inside the cube. Considering the number of face ambiguities, five distinct cases are possible:

Case 7-1: All positive grid points are separated and they are not joined inside the cube;

Case 7-2: Positive grid points on one ambiguous face are joined while others are separated;

Case 7-3: Positive grid points on two ambiguous faces are joined while others are separated;

Case 7-4-1: All positive grid points are joined on the three ambiguous faces; the negative grid point shared by three ambiguous faces is separated from other negative grid points;

Case 7-4-2: All positive grid points are joined on the three ambiguous faces; the negative grid point shared by three ambiguous faces is joined inside with other negative grid points.

Configuration 10: For these configurations two ambiguous faces need to be handled. By reversing the positive/negative signs, we are able to ensure that at least one of the ambiguous faces has separated positive grid points. Particularly, we rotate the cubes so that the positive grid points on the top face are separated. When the

Table 2 31 cases for 3D cubic cells with meshes generated using the modified DC. Each cell has a few positive grid points (red), several minimizers (blue) and some minimizers of adjacent cells (green).

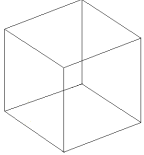
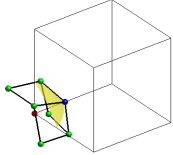
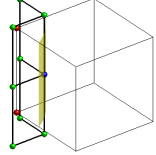
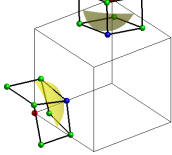
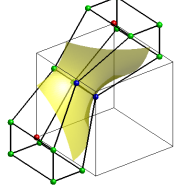
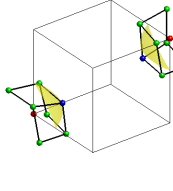
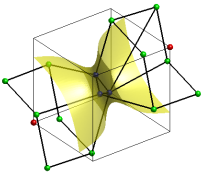
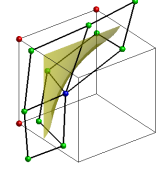
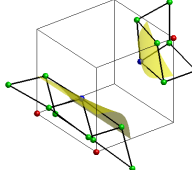
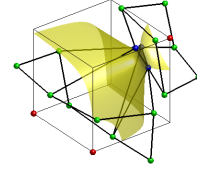
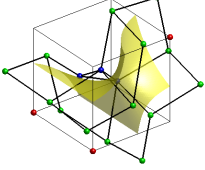
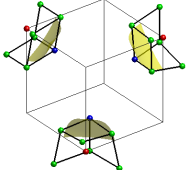
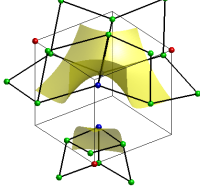
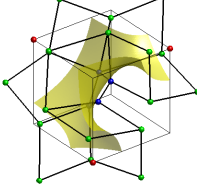
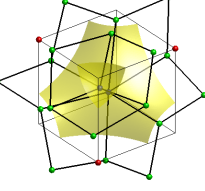
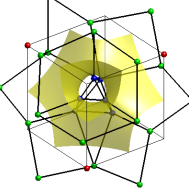
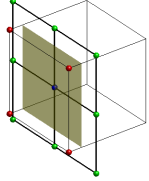
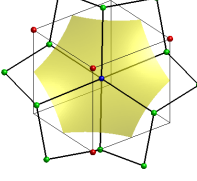
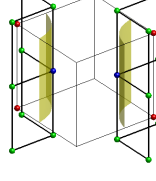
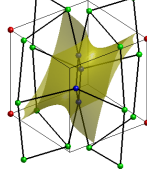
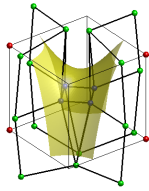
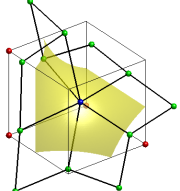
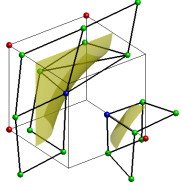
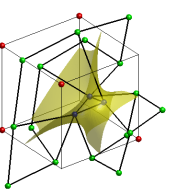
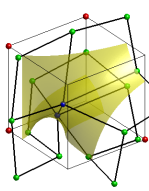
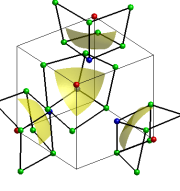
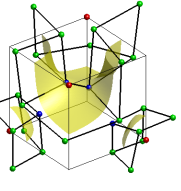
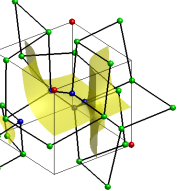
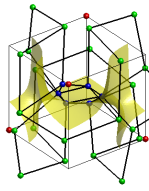
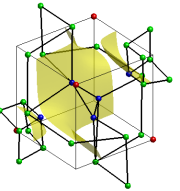
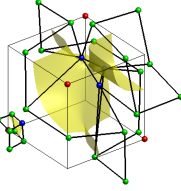
			
Case 0	Case 1	Case 2	Case 3-1
			
Case 3-2	Case 4-1-1	Case 4-1-2	Case 5
			
Case 6-1-1	Case 6-1-2	Case 6-2	Case 7-1
			
Case 7-2	Case 7-3	Case 7-4-1	Case 7-4-2
			
Case 8	Case 9	Case 10-1-1	Case 10-1-2

Table 2 (continued)

			
Case 10-2	Case 11	Case 12-1-1	Case 12-1-2
			
Case 12-2	Case 13-1	Case 13-2	Case 13-3
			
Case 13-4	Case 13-5-1	Case 13-5-2	

positive grid points on the two ambiguous faces are both separated, interior ambiguities may be introduced. There are three cases belonging to Configuration 10:

Case 10-1-1: Positive grid points on the top and the bottom faces are both separated; no positive grid points on the body diagonal are joined from inside.

Case 10-1-2: Positive grid points on the top and the bottom faces are both separated; however, two grid points on the body diagonal are joined.

Case 10-2: Positive grid points on the top face are separated; however, positive grid points on the bottom are not. In this case the iso-surface forms a typical saddle shape.

Configuration 12: There are two face ambiguities in this configuration. By reversing the positive/negative signs, we are able to ensure that the positive grid points are separated on at least one ambiguity face. Interior ambiguity arises when the positive points on both ambiguous faces are separated. There are a total of four cases.

Case 12-1-1: Positive grid points are separated on both ambiguous faces; no body diagonal consisting of two positive grid points is joined from inside.

Case 12-1-2: Positive grid points are separated on both ambiguous faces; one body diagonal consisting of two positive grid points is joined inside the cube.

Case 12-2: Positive grid points are joined in one face while separated in another face. Depending on which one is joined, a mirror case is also alike.

Configuration 13: This configuration is the most complex one since its six faces are all ambiguous. By reversing the positive/negative, we can always ensure that the number of faces with separated positive grid points is greater than or equal to the number of faces with joined positive grid points. When that number is three, additional interior ambiguities may also arise. These ambiguities lead to a total of six distinct cases.

Case 13-1: Positive grid points are separated on all six ambiguous faces.

Case 13-2: Positive grid points are separated on five ambiguous faces while joined on the other one.

Case 13-3: Positive grid points are joined on four ambiguous faces while separated on the other two. Due to the nature of the tri-linear interpolation function [4], these two other faces must be adjacent to each other.

The remaining cases all have three ambiguous faces with positive grid points joined. These three faces must share one common grid point due to the nature of the tri-linear interpolation function. Depending on whether the common grid point is positive, different cases are possible. If yes, interior ambiguity arises.

Case 13-4: The common grid point of the three joined faces is negative.

Case 13-5-1: The common grid point is positive, and it is separated from all other positive grid points.

Case 13-5-2: The common grid point is positive, and it is separated from other positive ones; all negative points are joined from inside. A complementary case is also possible when we reverse all the positive/negative signs. In that case, all positive grid points are joined inside the cell while all negative ones are separated.

All of these 31 cases are shown in Table 2. For each leaf cell in the octree, we categorize it into one of the cases above. Once the classification is made, we need to modify the dual mesh and incorporate the specific topologies.

5.2 Handling 3D Ambiguities in Triangulation

It is obvious that resolving 3D ambiguities is much more complicated, not only in terms of the number of cases but also in terms of the complexity of each case. To incorporate these features, we traverse all the octree leaf cells to insert minimizers and to re-connect the triangles.

5.2.1 Adding Minimizers

To represent correct topologies, we may need multiple minimizers within one cubic cell. In order to add appropriate number of minimizers we need to analyze the intersections between the cubic cell and the iso-surface. For each cell face, there are four types of intersection according to the tri-linear interpolation in Equation (2):

Type 0: The iso-surface does not intersect the cell face; and no minimizer is needed at all.

Type 1: The iso-surface intersects the cell face only once; and one minimizer should be inserted nearby. If one face is adjacent to another one which also needs new

minimizers (Type 1, Type 2), these two faces can share one common minimizer. The minimizers are located in the position where the QEF is minimized. In 2D, these minimizers are actually the intersections of the tangential lines to the isosurface.

Type 2: The iso-surface intersects the cell face twice; in this case we need at least two minimizers to approximate the iso-surface.

Type 3: The iso-surface cuts through the cell forming one “tunnel” inside; at least three minimizers are needed to approximate the interior structure.

5.2.2 Re-Connecting Triangles with New Minimizers

With enough minimizers we are able to re-connect them so that the resulting triangulation reflects the correct topology. Similar to 2D cases, we first (1) find all the quads sharing the original minimizer; and then (2) decide which sign change edges these quads associate to; in the following we (3) associate the sign change edge with the appropriate minimizer(s); in the next stage we (4) re-connect the quads or pentagons using the new minimizers; and finally (5) split the quads and pentagons to form a triangular mesh.

Among these five steps the third one is the most critical. After the association between the sign change edge and the minimizer is created, all quads associating to that sign change edge are connected to the very minimizer(s). To illustrate, we show an example depicting Case 3-2 with details, see Fig. 5.

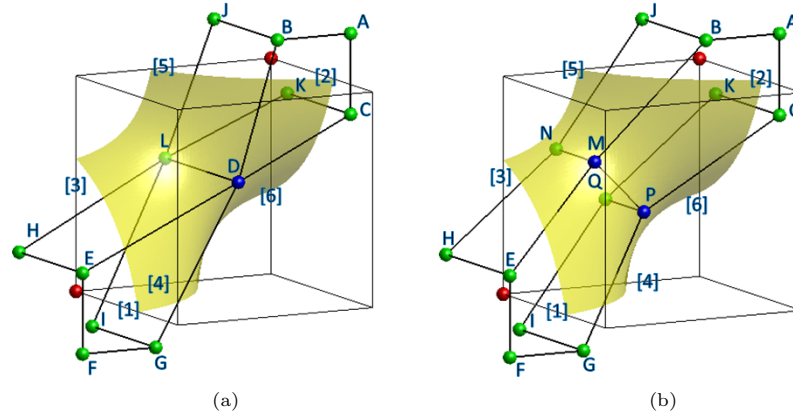


Fig. 5. The 3D re-meshing for Case 3-2. The numbers denote edge indices while the capital letters denote minimizers. The red points are positive grid points, the blue nodes are minimizers of this cell, and finally the green ones are minimizers of adjacent cells.

In the beginning we have a base mesh created using the standard DC, see Fig. 5(a). To incorporate the correct topologies we take the following two steps:

Step 1 (Insert multiple minimizers): We check the intersection between each cell face and the iso-surface. Only one intersection belongs to Type 2 (back face) while all the others belong to Type 0 or 1. Therefore two new minimizers (M and

P) are inserted near that face as shown in Fig. 5(b). Note that in the adjacent cell that shares the ambiguous face, similar techniques are carried out as well; therefore two new minimizers (N and Q) of the adjacent cell are also inserted.

Step 2 (Re-connect the mesh): We traverse the quads sharing the original minimizer, node D , find their sign change edges, and then associate each of them to one minimizer. At this point we have the following six configurations:

1. Quad \overline{DBJL} associates with the sign change edge [5], and it is associated with minimizer M, forming Quad \overline{MBJN} .
2. Quad \overline{DLHE} associates with the sign change edge [3], and it is associated with minimizer M, forming Quad \overline{MNHE} .
3. Quad \overline{DCKL} associates with the sign change edge [6], and it is associated with minimizer P, forming Quad \overline{PCKQ} .
4. Quad \overline{DLIG} associates with the sign change edge [4], and it is associated with minimizer P, forming Quad \overline{PQIG} .
5. Quad \overline{DCAB} associates with the sign change edge [2], and it is associated with two minimizers M and P at the same time, forming Pentagon \overline{MPCAB} .
6. Quad \overline{DEFG} associates with the sign change edge [1]. Similar to Quad \overline{DCAB} , it is also associated with P and M altogether, forming Pentagon \overline{MEFGP} .

By now we have traversed all the quads sharing the original minimizer D and re-constructed a hybrid mesh consisting of quads and pentagons. This hybrid mesh can be easily split into triangular meshes. Following the same procedure we are able to handle any of the 31 cases and incorporate correct topologies into the triangular mesh. Fig. 6 shows the meshes of five regions before and after re-connections. These figures show the topology Case 3-1, Case 3-2, Case 4-1-2, Case 7-4-2 and Case 10-2, respectively. The green lines or the blue node denote non-manifold situations, where one edge is shared by more than two triangles, or one vertex whose neighborhood is not like a disc. Note in (c), along the red dashed axes, an hollow tunnel topology is missing, and the our re-connection algorithm recovers it. In these five figures, it is obvious that the topology ambiguities are resolved and appropriate meshes are built.

Till now the mesh is topologically correct; however, it may have low-quality triangles that influence the convergence and stability of the numerical solutions. Therefore, we need to take an additional step to relocate these vertices as well as to improve the mesh quality.

To begin with, we need a metric to measure the quality of a triangle. Here, we choose the aspect ratio, defined as $2 * r_{in}/r_{out}$, where r_{in} is the radius of the inscribed circle, and r_{out} is the radius of the circumcircle. A perfect triangle has the aspect ratio of 1; therefore our goal is to increase the ratio toward 1. To attain this goal, a Laplacian smoothing technique is applied. Laplacian smoothing is a fast and efficient method to smooth a triangular mesh. For each mesh vertex, Laplacian smoothing relocates its position to the weighted average of the nodes connecting to it.

6 Results and Discussion

The developed algorithm is automatic and robust for meshing domains with topology ambiguities. We have applied our algorithm to three complicated datasets (Figs. 1, 7, 8). Our technique generates quality meshes with ambiguities resolved and feature

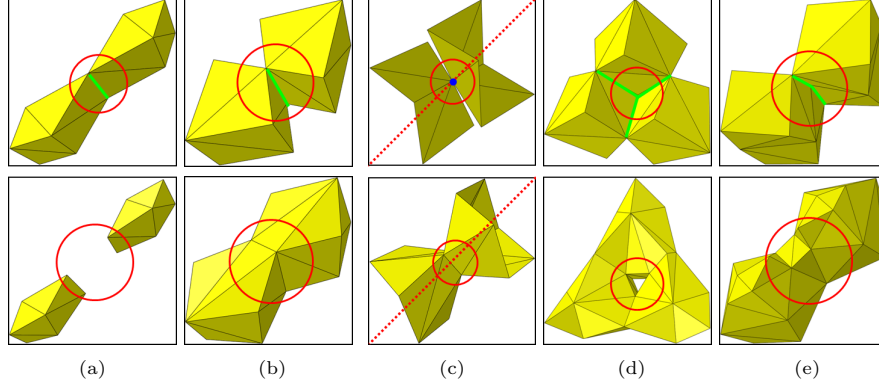


Fig. 6. Meshes of five regions before (first row) and after (second row) the reconnection. The red circles indicate the locations of the ambiguity, green lines or the blue dot indicate non-manifold situations, and red dashed lines denote the axes of a tunnel topology. (a) Case 3-1; (b) Case 3-2; (c) Case 4-1-2; (d) Case 7-4-2; and (e) Case 10-2 examples.

preserved. The computations were based on a PC with Intel Q9600 CPU and 4GB DDR-II memories. Table 3 shows statistics of the generated triangular meshes. The mesh size is increased after resolving the ambiguities, since one leaf cell may contain multiple minimizers. The mesh quality is improved greatly after smoothing. Note that the computation time includes the time for the base mesh generation, topology preservation, and the following smoothing. The smoothing process takes most of the time and has an approximately linear running time.

Our technique is a combination of DC and MC. When we generate the base mesh, we analyze the sign change edges and construct a dual of the octree; when we resolve and approximate the ambiguities, we traverse all the octree leaf cells to re-connect the mesh. In this manner we have the advantages of both DC (can handle large datasets with adaptive octree; better qualities) and MC (can preserve topology ambiguities). Furthermore, to make meshes conformal across cell boundaries, the remeshing templates should be carefully designed so that no discrepancies or gaps exist on both sides of one cell face. Meanwhile, only face ambiguities influence the meshes in other cells. As a result, if all face ambiguities are handled consistently, as what we have done in our algorithm, we will obtain conformal meshes everywhere.

7 Conclusion

We have developed an efficient and automatic method to generate triangular iso-surfaces with correct topology. We begin from the standard DC method and generate one mesh without considering topology ambiguities. To resolve these ambiguities, we first model the interior regions with a tri-linear interpolation, and then compute the face and body saddle points, and then discuss all the 31 possible configurations. After categorizing all the cells into different topology groups, we traverse these cells and re-connect the triangles according to a set of pre-defined templates. In the end we relocate the vertex positions and thus improve the mesh quality. We have already

Table 3 Statistics of the generated meshes

Model	Mesh	Mesh Size (Vert #, Elem #)	Aspect Ratio (Worst, Best)	Time (s)	Ambiguity Cell #
Trabecular Bone 1 (Fig. 1)	Original	(98,694, 197,636)	(0.0006 1.0)		
	Smoothed	(98,944, 197,884)	(0.158, 1.0)	28	887
Trabecular Bone 2 (Fig. 7)	Original	(134,925, 271,224)	(0.000066, 1.0)		
	Smoothed	(135,911, 272,150)	(0.168, 1.0)	36	241
Foam (Fig. 8)	Original	(186,629, 374,420)	(0.030 1.0)		
	Smoothed	(187,046, 374,824)	(0.16, 1.0)	51	388

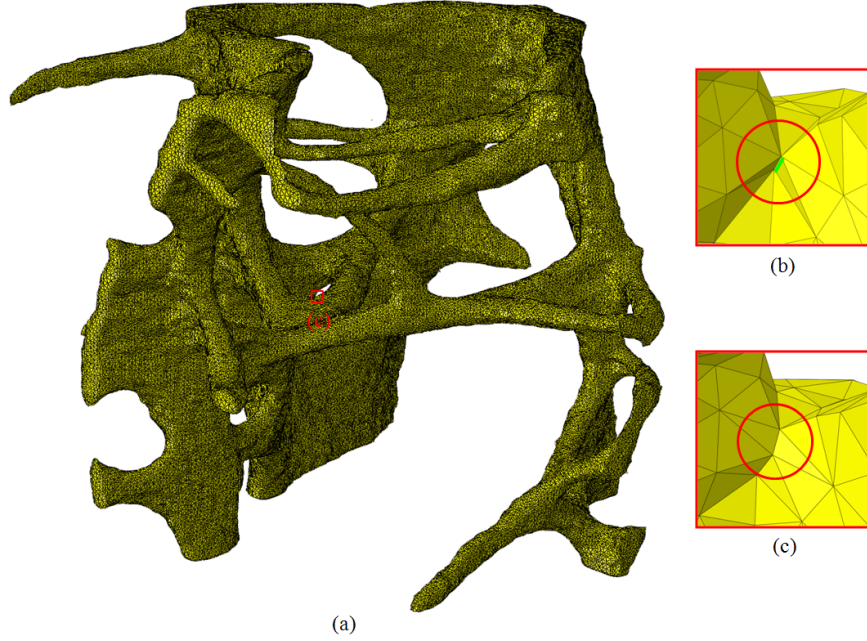


Fig. 7. (a) Triangular mesh of another trabecular bone structure. (b) and (c) show zoom-in details of a local region with topology ambiguity (see the red circle). The green line in (b) denotes one non-manifold edge, which is resolved in (c).

applied our approach to three complicated datasets and obtained good results. In the future we will extend our algorithm to domains with multiple components. Instead of continuous function values, these data grids have distinct component IDs. Since one cell may have up to eight components, the ambiguity resolving is much more complicated and deserves great attention.

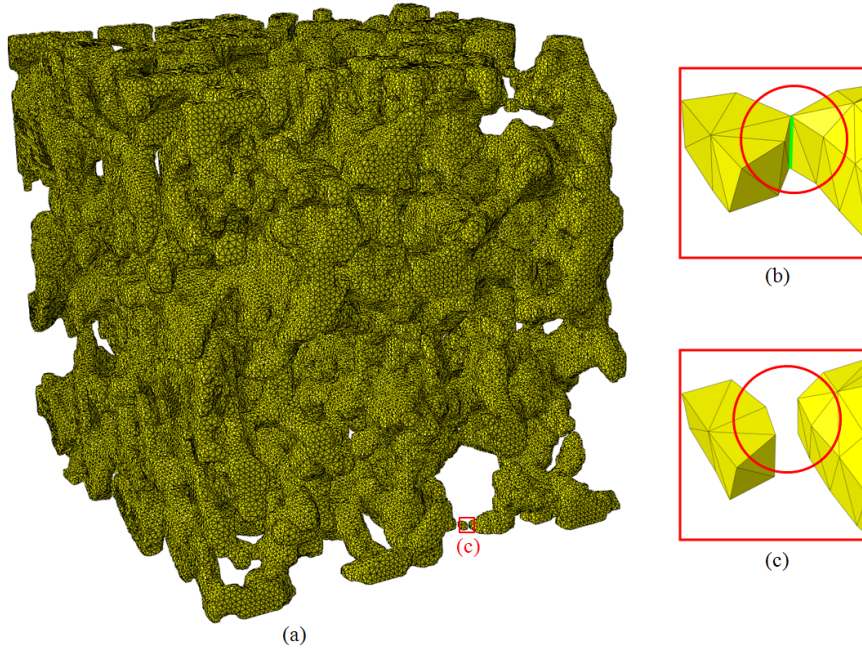


Fig. 8. (a) Triangular mesh of a foam material. (b) and (c) show zoom-in details of a local region with topology ambiguity (see the red circle). The green line in (b) denotes one non-manifold edge, which is resolved in (c).

Acknowledgement

We would like to thank W. Wang for useful discussions on topology categorization. The research was supported in part by a NSF/DoD-MRSEC seed grant.

References

1. Ashida K, Badler N (2003) Feature preserving manifold mesh from an octree. The 8th ACM Symposium on Solid Modelling and Applications'03, pp. 292–297
2. Bajaj C, Pascucci V, Schikore D. (1996) Fast isocontouring for improved interactivity. IEEE Symposium on Volume Visualization'96, pp 39–46
3. Brodsky D, Watson B (2000) Model simplification through refinement. Graphics Interface, pp. 221–228
4. Chernyaev E (1995) Marching cubes 33: Construction of topologically correct isosurfaces. Technical Report CH/95-17, CERN
5. Durst M (1988) Letters: Additional reference to marching cubes. Computer Graphics, 22: 72–73
6. Garland M, Heckbert P (1988) Simplifying surfaces with color and texture using quadric error metrics. IEEE Visualization'88, pp. 263–269
7. Garland M, Shaffer E (2002) A multiphase approach to efficient surface simplification. IEEE Visualization'02, pp. 117–124

8. George PL, Borouchaki H (1998) Delaunay triangulation and meshing, application to finite elements. Hermes Science Publisher
9. Nielson G (2004) Dual marching cubes. IEEE Visualization'04, pp. 489–496
10. Ju T, Losasso F, Schaefer S, Warren J (2002) Dual contouring of Hermite data. ACM Transactions on Graphics, 21: 339–346
11. Kanaya T, Teshima Y, Kobori K, Nishio K (2005) A topology-preserving polygonal simplification using vertex clustering. GRAPHITE, pp. 117–120
12. Kobbelt L, Botsch M, Schwaner U, Seidel H (2001) Feature sensitive surface extraction from volume data. SIGGRAPH'01, pp. 57–66
13. Lopes A, Brodlie K (2003) Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing. IEEE Transactions on Visualization and Computer Graphics, 9(1): 16–29
14. Lorensen W, Cline H (1987) Marching cubes: A high resolution 3D surface construction algorithm. Computer Graphics, 21: 163–169
15. Natarajan B (1994) On generating topologically correct isosurfaces from uniform samples. The Visual Computer, 11: 52–62
16. Nielson G, Hamann (1992) The asymptotic decider: Resolving the ambiguity in marching cubes. IEEE Visualization'92, pp. 83–91
17. Schaefer S, Ju T, Warren J (2007) Manifold dual contouring. IEEE Transactions on Visualization and Computer Graphics, 13(3): 610–619
18. Schaefer S, Warren J (2004) Dual marching cubes: Primal contouring of dual grids. Computer Graphics and Applications'04, pp. 70–76
19. Varadhan G, Krishnan S, Kim Y, Manocha D (2003) Feature-sensitive subdivision and iso-surface reconstruction. IEEE Visualization'03, pp. 99–106
20. Westermann JR, Kobbelt L, Ertl T (1999) Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. The Visual Computer, 15(2):100–111
21. Wilhelm J, Van Gelder A. (1992) Octrees for faster isosurface generation. ACM Transactions on Graphics, pp. 57–62.
22. Zhang N, Hong W, Kaufman A (2004). Dual contouring with topology preserving simplification using enhanced cell representation. IEEE Visualization'04, pp. 505–512
23. Zhang Y, Bajaj C (2006). Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. Computer Methods in Applied Mechanics and Engineering, 195(9-12): 942–960
24. Zhang Y, Bajaj C, Sohn B (2005) 3D finite element meshing from imaging data. Computer Methods in Applied Mechanics and Engineering, 194: 5083–5106
25. Zhang Y, Bajaj C, Xu G (2009). Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. Communications in Numerical Methods in Engineering, 25(1): 1–18
26. Zhang Y, Hughes T, Bajaj C (2010). An automatic 3D mesh generation method for domains with multiple materials. Computer Methods in Applied Mechanics and Engineering, 199(5-8): 405–415